

# FEATURE EXTRACTION FOR POST-TEST DIAGNOSTICS

June F. Zakrajsek<sup>1</sup>, Christopher E. Fulton<sup>2</sup>, Claudia M. Meyer<sup>3</sup>

<sup>1</sup>NASA Lewis Research Center  
21000 Brookpark Road  
Cleveland, Ohio 44135

<sup>2</sup>Analex Corporation  
3001 Aerospace Parkway  
Brookpark, Ohio 44142

<sup>3</sup>NYMA, Inc.  
2001 Aerospace Parkway  
Brookpark, Ohio 44142

## Abstract

Feature Extraction algorithms, which identify prominent events of a sensor signal trace, have been developed by the NASA Lewis Research Center. Feature Extraction algorithms are used to automate the data review process used to assess the condition of propulsion systems. These algorithms have been encoded and their versatility has been demonstrated on Space Shuttle Main Engine and Atlas/Centaur pneumatic and electrical subsystem data. Detailed logic and developmental issues for each of the primary Feature Extraction algorithms are given. These algorithms detect peaks, drifts, spikes, level shifts, erratic and noisy behavior, limit exceedances, and signal start bias.

## I. Introduction

The Post Test Diagnostic System (PTDS) was developed to reduce analysis time and to increase accuracy and repeatability of rocket engine test fire data analysis by automatically retrieving data, extracting predefined events and analyzing these events based on heuristics provided by expert analysts.<sup>1,2,3</sup> In order to fully process the data, an event based analysis (i.e., analysis based on features within a single signal or set of signals) was incorporated.

Feature extraction routines are a critical part of the PTDS analysis process since they provide the mechanism which fully reduces the data to specific events. These events are further processed using heuristic, model, and case-based reasoning techniques to determine the condition of the system.<sup>4</sup> The feature extraction routines have been designed to be modular so that they can be easily applied to various vehicle subsystems.

The feature extraction module, developed for the PTDS, contains algorithms which can be separated into two categories. The first category,

called primary algorithms, analyzes data from single sensors, while the second category analyzes data from multiple sensor sets. The second category routines are typically combinations of the primary algorithms and are often very specific to the system being analyzed. The focus of this paper will be on the algorithms which fall into the first category, primary feature extraction algorithms: peaks, drifts, spikes, level shifts, limit exceedances, erratic and noisy behavior, and signal start biases.

The paper will discuss the implementation issues for feature extraction, describe each of the algorithms and their purpose, and discuss feature extraction development and application issues as they pertain to data from the Space Shuttle Main Engine (SSME) and from General Dynamics Centaur pneumatic and electrical subsystems.

## II. Implementation of Feature Extraction Algorithms

Rocket engine data analysts perform the same basic analysis steps regardless of the system.<sup>5,6,7</sup> The analysts identify events contained within a sensor trace and apply their system knowledge to understand the meaning of these events. The PTDS and the feature extraction algorithms were designed to facilitate the automation of this data analysis procedure. In addition, the design methodology separates all system dependent variables from algorithm logic to facilitate application to different systems.

The routines are written in 'C' and the code is implemented such that algorithm dependent parameters can be changed without code modifications.<sup>8</sup> In limited cases, the algorithms can be used from within the CLIPS expert system shell via the CLIPS command line.<sup>9</sup>

A commands table, which is a file or a database table, contains features to be extracted from indicated parameters, over a given time period, using

several different feature attributes. All of this information is application specific, and will change from system to system. The feature attributes include both statistical and parameter or system based limits, such as historical based means and standard deviations for signal start bias, and hardware based redline limits for limit exceedance.

In addition to the information contained in the command file and the database, the algorithms require time stamped data. Output from the algorithms include the test descriptor, the sensor name, feature start and end times and a measure of the feature's magnitude. Other feature characteristics are included whenever they are needed to further define the feature. Output is made to either a set of relational database tables, or to an ASCII flat file.

### III. Feature Extraction Algorithms Description

The feature algorithms are described below. Each of these algorithms has been successfully applied to Launch Vehicle data. Proper setting of the feature attributes is critical to the correct identification of the features. These attributes can vary greatly depending on the system being monitored. Some features can be detected with more than one algorithm, all algorithms that have been implemented are discussed.

For the drift and the slope based level shift and peak algorithms the data can be smoothed in order to reduce random variations and improve detection quality. The feature attributes for these algorithms contain a smoothing variable, which sets the smooth window size. A value of one prevents smoothing. The smoothing algorithm replaces each data point with the average of the data points contained within the smooth window, which is centered about the point.

The drift, y-intercept based level shift and noisy behavior algorithms use either averaged data or standard deviations which are calculated over defined time segments during data loading. The defined segments are typically twenty-five to fifty data points wide. The average and standard deviation of the data represent the signal's behavior during the data segment.

The nomenclature for the features is arbitrary, but does indicate the general characteristics of the desired event. Figures 1 through 8 provide visual illustrations of each feature type found in either SSME or Centaur pneumatic or electrical system data.

### Level Shifts

Level shifts are significant rapid changes in average sensor magnitude where the average values before and after the event remain constant for a predefined time. Level shifts can be detected with one of two algorithms, depending on the nature of the data. The first algorithm is based on changes in y-intercept and should be used over data intervals where the data variation is small compared to the overall data range. The second algorithm is based on changes in slope and can be used on data where the data variation relative to the overall data range is large. The level shift algorithm based on monitoring the y-intercept requires a minimum amplitude attribute. The slope based algorithm also needs attribute values for slope standard deviation and sensor signal standard deviation, the number of points per line fit and the smoothing window size. An example of two consecutive level shifts is shown in Figure 1.

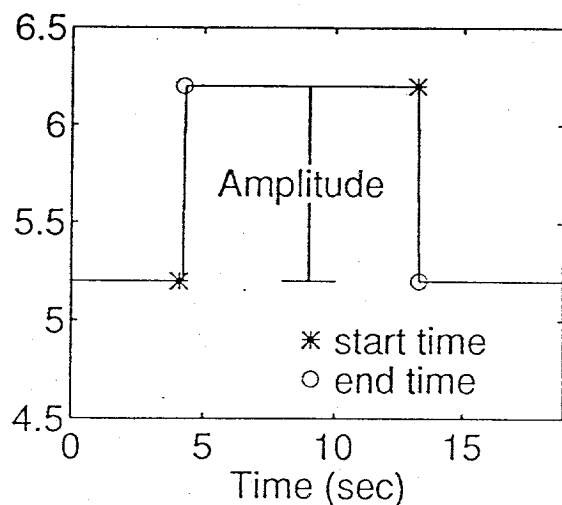


Figure 1: Example of consecutive level shifts.

The y-intercept based algorithm divides the signal into non-overlapping intervals and performs a first order fit over each interval. The average and three standard deviation values for the y-intercept terms are then calculated for use in level shift detection. A level shift is indicated when the y-intercept deviates significantly from the average of the y-intercept terms. The level shift amplitude is calculated from the difference in the predicted values at the beginning and end of the level shift. This amplitude is then tested against the minimum amplitude to determine if a level shift should be

declared.

The second algorithm also breaks the signal into non-overlapping intervals and performs a least squares line fit over each smoothed interval. The slopes of each line segment are then tested against three times a slope standard deviation attribute to find places of exceedance. The first occurrence of an exceedance determines the beginning of a period of interest. The algorithm then monitors the slope to determine when the interval of interest has ended. This occurs when the slope has returned to a statistical zero. Once this interval has been defined, the amplitude of the level shift is calculated from the difference of the predicted values at the interval stop and start times. If this amplitude is greater than three times the standard deviation of the smoothed signal, a level shift is declared. The standard deviation attribute is based on historical nominal data.

### Peaks

The purpose of the peak feature extraction routine is to detect areas of significant positive or negative data excursions where: the starting and ending data magnitudes are reasonably equal, the duration of the excursion is within a user defined time interval, and the magnitude is greater than a predefined height. Figure 2 shows a peak.

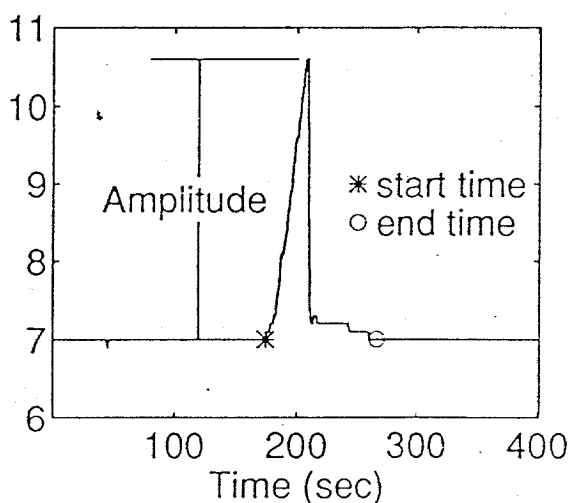


Figure 2: Example of peak feature.

Peaks may be detected using either of two algorithms. The first algorithm uses attributes for minimum height and minimum width. This

algorithm computes slopes using averaged data. The slope is computed from the sensor's amplitude and corresponding time stamp values over the window being considered. Results from the slope computations are monitored for significant non-zero values. Once a significant slope is detected the onset time is recorded and the slope is monitored for a change in sign. When the slope changes sign, the peak has reached its maximum height, which is computed from the difference in signal values at the peak onset and peak height times. Monitoring of the slope continues until it returns to a near-zero state where the peak end time is noted. If the peak duration, end time - onset time, is less than the predefined minimum width, the peak is rejected. In addition, peaks which are less than the minimum height are also rejected.

After a peak area has been established, the signal during that interval is fitted using either a Gaussian model or "fast rise with exponential fall off" model to calculate additional peak characteristic values, such as the times where the peak is at half height and the corresponding magnitudes. This information, as well as the peak magnitude, the time of maximum magnitude, and the peak start and end times are then recorded in a database table or output file.

The second peak algorithm is computed concurrently with the slope based level shift algorithm previously described, and therefore uses the same attributes of slope and sensor signal standard deviations, the number of points per line fit and smoothing window size. In addition, the slope based peak algorithm uses a standard deviation multiplier attribute. In this algorithm, smoothed data values are divided into intervals whose size is defined by the number of points per line fit attribute. A first order curve fit is performed in each interval to determine the slope. An interval of interest is indicated by significant deviations in the slope from zero. An interval which has similar starting and ending magnitudes is a candidate for a peak. The peak height is then calculated as the difference between the maximum and minimum data points present during the peak interval. The height is checked against a minimum value which is the product of the signal standard deviation attribute and the multiplier attribute. The peak is rejected whenever the height is below the calculated minimum.

### Drifts

The purpose of the drift feature extraction algorithm is to detect positive or negative linear trends in sensor data. This algorithm requires the

definition of minimum and maximum slope attributes. A drift is shown in Figure 3.

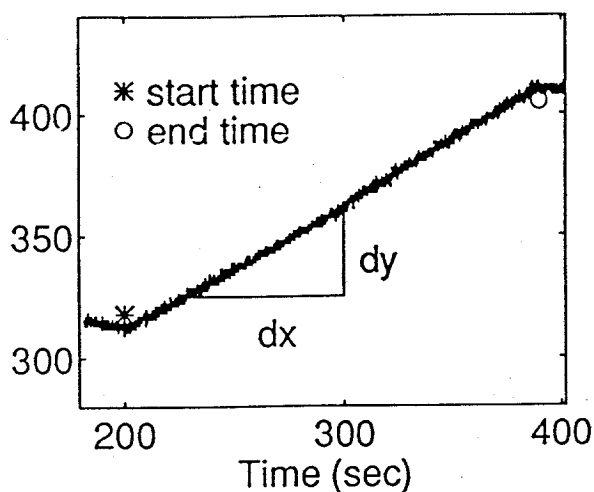


Figure 3: Example of drift feature.

The drift algorithm uses smoothed data that is recursively subdivided into periods of linear behavior. This is accomplished by adjusting the data within the interval under consideration so that both endpoints are equal to zero. The maximum magnitude within the interval is then checked to determine if it is greater than four times the average standard deviation of the signal. A magnitude greater than four average standard deviations indicates that the interval can be subdivided further. Once the periods of linear behavior have been determined, the sensor's amplitude and corresponding time stamp values at the endpoints of each period are used to calculate an average slope. This slope is checked against the minimum and maximum slope attributes. Slopes greater than the first threshold indicate the sensor is drifting. Slopes greater than the second threshold are rejected so as to minimize overlap with features detected by the level shift algorithm.

#### Spikes

The spike feature extraction routines determine if sensor data exhibits singular large excursions within a short time span. The spike algorithm uses attributes for maximum width, minimum height, and a standard deviation multiplier. An example of a spike is shown in Figure 4.

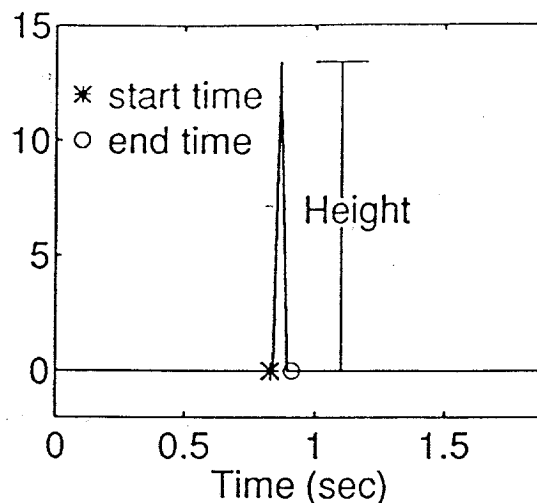


Figure 4: Example of a spike feature.

The spike algorithm divides the data into windows of approximately equal length. These windows can be overlapped so that spikes can be found even when they occur at the edges of sample windows. The data within each window is fit with first and second order polynomials. Chi-square statistics are calculated for each curve fit to obtain a measurement of the goodness of each fit. The chi-squared values are compared and the better of the two curve fits is selected. The actual data values are then compared to the chosen curve fit and a standard deviation is calculated for the residuals. The deltas between a current point and the three previous points in the interval are compared to either the standard deviation of the residuals multiplied by the standard deviation multiplier attribute or the minimum height attribute, whichever is greater. The minimum spike height is usually defined in terms of a multiplier times the least significant bit of the analog-to-digital level. A spike is identified when the delta is greater than the selected threshold and the spike duration is less than a user defined spike width.

#### Erratic Behavior

The erratic behavior routine detects slow time varying data. This algorithm requires the setting of the expected standard deviation attribute and its multiplier. An example of this feature is shown in Figure 5.



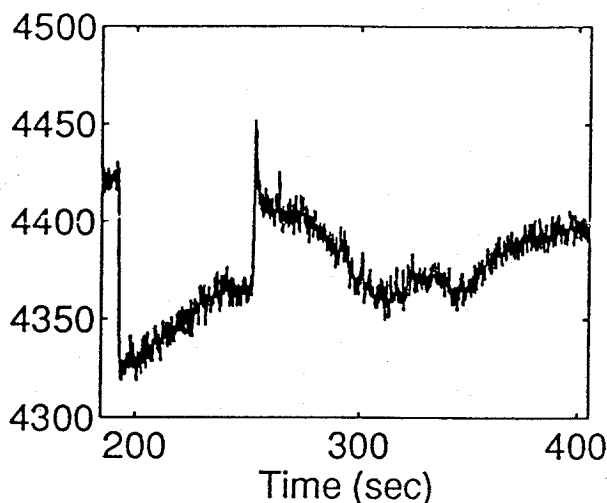


Figure 5: Example of erratic behavior feature.

A first and second order fit of the test data is made and the better of the two fits is chosen based on the chi-square statistic. The standard deviation of the residuals from the selected fit is then calculated. This standard deviation is compared to a minimum threshold which is the product of the expected standard deviation and its multiplier attribute. The sensor is declared erratic when the standard deviation is greater than the calculated minimum threshold.

#### Noisy Behavior

This algorithm is used to detect sensors which are exhibiting larger than expected signal variations. Noisy behavior is usually indicative of an instrumentation problem. The modifiable attribute for Noisy Behavior is the maximum allowable standard deviation. An example of noisy behavior can be seen in Figure 6.

The noise feature extraction algorithm uses the standard deviation values which are calculated over small time segments for each sensor upon data loading. A comparison of each standard deviation is made against the maximum allowable standard deviation. The maximum limit is typically based on historical data from nominal sensors. The sensor is declared noisy when a calculated standard deviation is greater than the predefined limit.

#### Limit Exceedance

The limit exceedance routine checks sensor

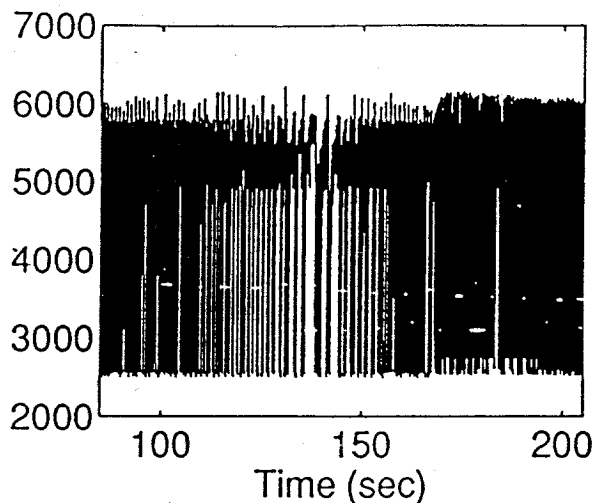


Figure 6: Example of noisy behavior feature.

data over specified time intervals to determine if they exceed upper and/or lower limits. These limits include reasonableness limits for sensor qualification and redline checks based on hardware considerations. Modifiable attributes are minimum exceedance duration time and the upper and lower limits. An example of this feature is shown in Figure 7 where the redline limit for the SSME parameter was exceeded at approximately 770 seconds.

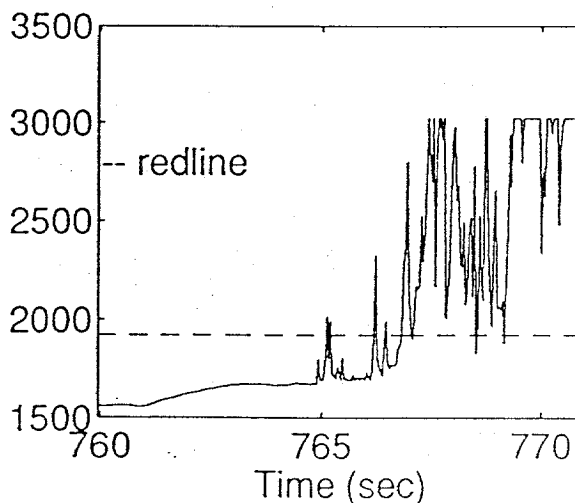


Figure 7: Example of limit exceedance feature. For this example, an SSME parameter exceeded its redline.

An exceedance is detected by performing a point-by-point comparison of the sensor data to the applicable limit. The duration of the limit exceedance is compared to the user defined minimum and the exceedance is rejected when the duration is less than the minimum.

#### Signal Start Bias

The purpose of the signal start bias routine is to determine whether a sensor is within acceptable limits just prior to system start. User modifiable attributes are the minimum and maximum average values for a sensor during pre-start. Figure 8 shows a typical start bias for an SSME actuator position and a suspect low start bias for the current test.

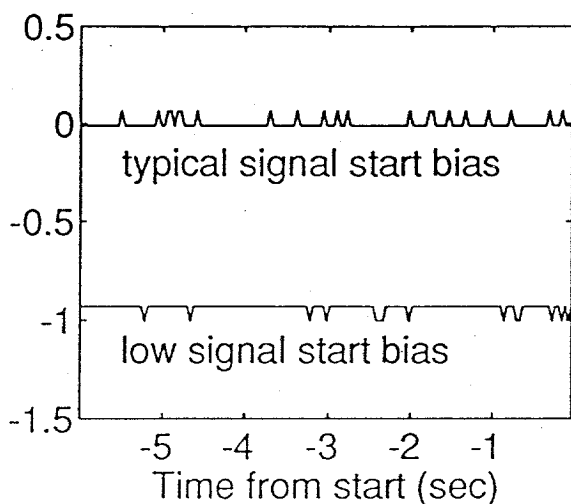


Figure 8: Example of start signal bias feature.  
Typical start level for an actuator is zero, for this test start level was low.

Each sensor is averaged over a short time interval just prior to system start. The time interval must be kept large enough to obtain a statistically meaningful average value but small enough to eliminate data responding to scheduled system events. The average is compared against the maximum and minimum limits. The sensor is declared out of tolerance when it is not within the range defined by these limits. Typically, these limits are defined as the maximum and minimum averages of the sensor on previous tests where the sensor was not declared faulty by the expert analysts.

#### IV. Feature Extraction Algorithm Development and Application Issues

In developing the feature extraction algorithms, two major areas need to be addressed. These are accurate identification of features and minimum computation time. The latter is mostly an encoding and platform issue while the former impacts the logic used. Additionally, there are several application issues which impact the accurate identification of features: transformation of features from the visual to numerical domain, data variation and threshold determination, and flexibility.

#### Data Transformation

Transformation of features from a visual domain to a numerical domain requires that an exact definition of the feature be known as well as all limits associated with the feature. These metrics are application and expert dependent. In order to judge a feature and its importance to the system diagnosis, an expert uses his or her knowledge about the system and the data. The knowledge about the data is usually acquired through the detailed and repeated study of multiple data plots. These plots are typically presented by using a y-scale which has been automatically scaled to accommodate all of the data within the data set. Therefore, the knowledge that experts have of features existing within the data has been biased by the y-axis range encountered in their experience base. The feature extraction algorithms must be able to account for this bias in order to correctly identify the features. For example, Figure 9 shows an SSME parameter which contains a drift at approximately 450 seconds. Due to the scale on the y-axis, the drift appears to have an 18 degree inclination. However, mathematically, the inclination is actually only 2 degrees. In order to correctly account for the y-range bias effect, the data and thresholds are normalized.

Finally, feedback between the expert and knowledge engineer is critical in order to map features identified by the expert to the feature extraction algorithms. For example, the spike seen in Figure 4 is commonly referred to as a 'glitch' by Centaur electrical system analysts. Another example is shown in Figure 10. This figure shows a Centaur pneumatic parameter which contains several level shift features found within the boxed portion of the signal. To the Centaur expert all of the features in Figure 10 are classified as level shifts, even though the longer duration level shifts could possibly be classified as drifts. While both examples show that either algorithm can be extended to detect features with similar attributes, ultimately it is the knowledge engineer's task to ensure the routines correctly

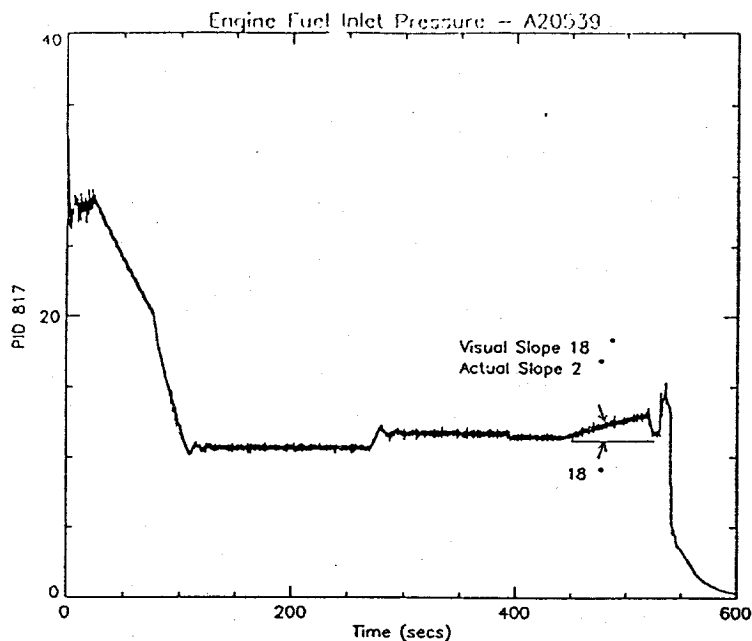


Figure 9: SSME parameter containing a drift feature, where the visual slope angle is  $18^\circ$  and the mathematical slope angle is  $2^\circ$ .

identify the features the analyst needs.

#### Data Variation

Another feature extraction issue is the ability to handle and correctly extract events from various data which cover a wide range of behavior. For instance, an application may require the entire range of the sensor be used. Such an example can be seen in Figures 10 and 11 which show signals with high variability. In Figure 10, the Centaur pneumatic system parameter shown has an overall range which is large compared to the range of the features which need to be detected between seven thousand and eleven thousand seconds. Figure 11 shows an SSME parameter, which exhibits a large range and a level shift prior to 100 seconds.

There are two ways to accommodate features that are small with respect to the overall range of a parameter. The first is to break down the data into smaller intervals which are nominally stationary. SSME data are preprocessed into periods where the commanded thrust level of the engine is constant. The test firing shown in Figure 11 contains eight levels of constant thrust, denoted by the numbers 1 through 8, which are centered over each period. The separation of the data into periods of constant thrust mimics the way the data analysts partition data during their post firing review.

The second option is to change the method

of determining the feature. For example, level shifts can be determined by using either a y-intercept method or a slope method. Both of these routines use similar logic, but, each monitors a different coefficient from the linear curve fit. The y-intercept method is better suited for those applications where the data range is small, and the slope method is better for data with a large range.

#### Algorithm Flexibility

The feature extraction algorithms have been developed to be easily applied to time stamped data generated by any type of system. In order to accomplish this, the algorithms accommodate system dependent changes in the analysis by explicitly setting the feature attributes. These attributes are defined either in the commands table or in the 'C' code. Changes made to the commands table do not require recompiling the feature extraction code. Where attribute changes require recompilation, the feature extraction logic has been developed to be insensitive to these changes. This is accomplished by separating all of the feature attributes into 'C' header files where changes to the code can be made without modifying the algorithm logic. This also allows the developer to easily find and maintain changes to the feature extraction application.

Some of the feature attributes require defining acceptable confidence intervals. Typically these thresholds are generated based on statistics from several past nominal data sets. This requires

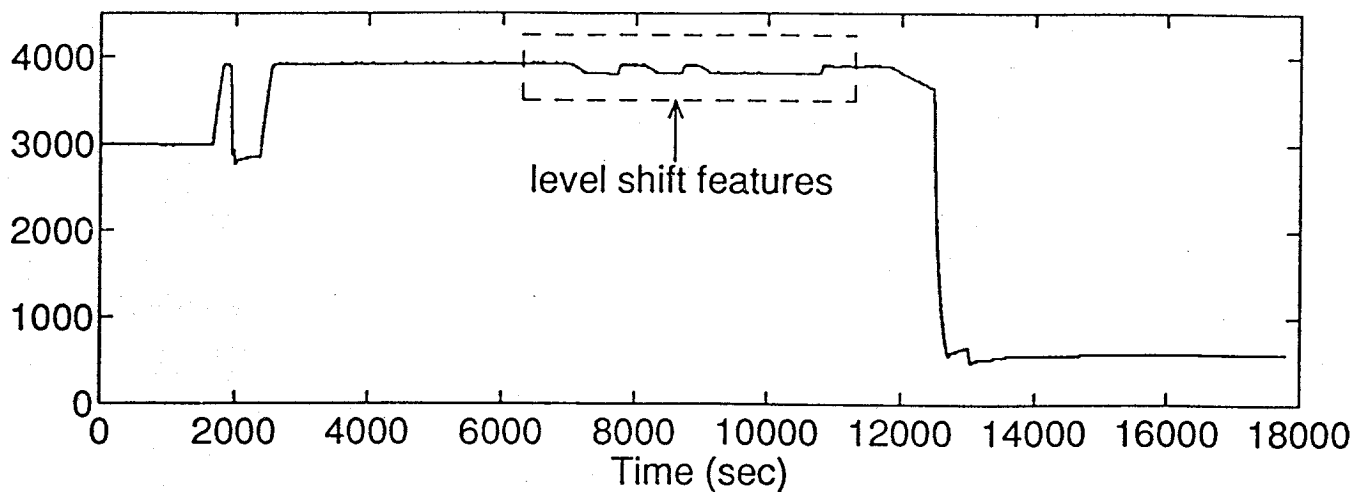


Figure 10: Centaur pneumatic data with a large overall range compared to the amplitude of the level shift features.

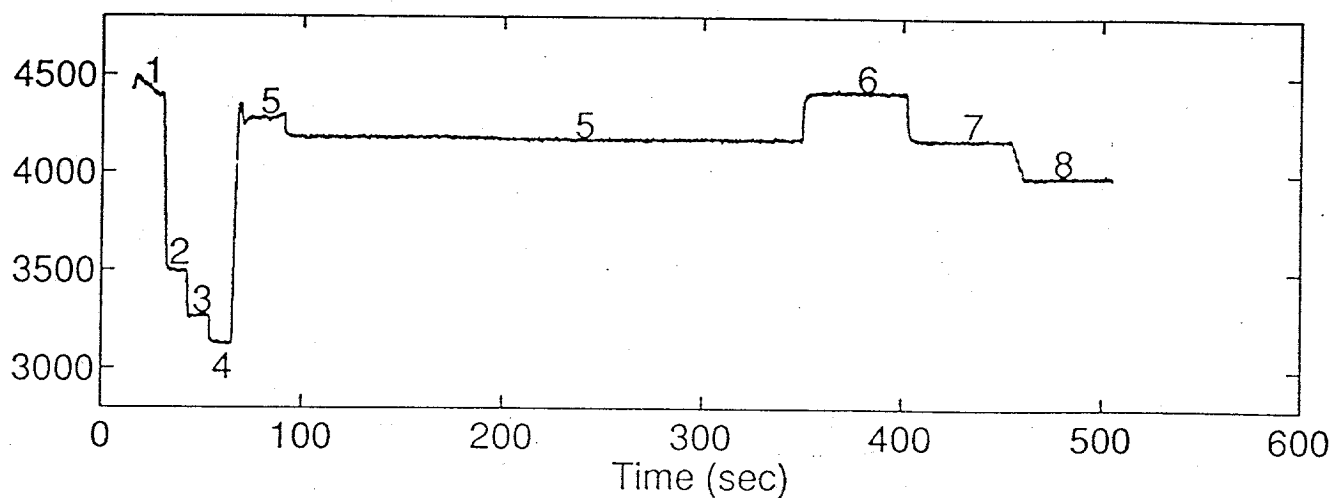


Figure 11: SSME parameter with a large overall range, which has been broken into periods of constant thrust denoted by the numbers 1 through 8.

a sufficient database of nominal data and the processing of this data. However, thresholds can also be set initially based on expert advice.

#### V. Summary

Automating post test and flight operations is directly impacted by the accurate and fast review of enormous amounts of data. An easily adaptable Post Test Diagnostic System has been developed to automate the data analysis process. The process of data analysis is similar regardless of the system being analyzed. This basic concept is being demonstrated by the development and application of

generic PTDS modules to a variety of vehicle subsystems.

Feature Extraction is critical to the data review process and the Post Test Diagnostic System. The primary Feature Extraction algorithms identify peaks, drifts, spikes, level shifts, noisy behavior, limit exceedance and signal start bias. These algorithms have been encoded using 'C' which can be embedded within any diagnostic system. The code developed for these algorithms has been designed to easily accommodate changes from different systems. This has been accomplished by separating all feature attributes and thresholds into either a commands table or 'C' header files.

There are several developmental issues which have been addressed. In many cases, the burden lies upon the knowledge engineer to correctly map features identified by the expert to the feature extraction algorithms. This task is directly influenced by transformation issues between visual and numerical representations, data variation, and data preprocessing requirements.

The Feature Extraction algorithms have been designed to transfer easily across applications. Examples have been given to demonstrate the ability of the Feature Extraction algorithms to successfully identify events within data taken from the Space Shuttle Main Engine and the Atlas/Centaur pneumatic and electrical subsystems.

#### VI. References

- (1) Dr. Pamela Surko, June F. Zakrajsek, PTDS: Space Shuttle Main Engine Post Test Diagnostic Expert System for Turbopump Condition Monitoring, SAE Aerotech '92, SAE Report 922059, October 1992.
- (2) June F. Zakrajsek, The Development of a Post-Test Diagnostic System for Rocket Engines, AIAA-91-2528. June, 1991.
- (3) Dr. Pamela Surko, Reusable Rocket Engine Turbopump Health Management System, Contract NAS3-25882.
- (4) T.W. Bickmore, W.A. Maul, A Qualitative Approach To Systemic Diagnosis of the SSME, 31st Aerospace Sciences Meeting & Exhibit, AIAA 93-0405, January 1993.
- (5) Cannon, I.D., Life Prediction for Rocket Engines Task 2: SSME Diagnostic System Final Report, Contract NAS3-25884, 1990.
- (6) Gage, M., DeHoff, R., Development of Life

Prediction Capabilities for Liquid Propellant Rocket Engines Task 4: Diagnostic System Architecture Study Final Report, CR 187112, 1990

- (7) Johnson, J.G., Schneemann, G.K., Automated Data Analysis for Space Launch Vehicles, AIAA 90-1312-CP, July 1990.
- (8) W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, "Numerical Recipes in C", Cambridge University Press, Cambridge, MA, 1992.
- (9) T.W. Bickmore, SSME HPOTP Post-Test Diagnostic System Enhancement Project Final Report, Task 11 of Contract NAS3-25883, February 1994.